Analysis of the representation of hand sketch neural network classifiers using simple Bézier curves

1st Bodică Septimiu-Călin
Babeș-Bolyai University
Faculty of Mathematics & Computer science
Cluj-Napoca
tbodica@gmail.com

***Abstract*—Sketch recognition differs from traditional photograph classification in that sketches are less visually complex, have less channels, and can usually be described as a sparse grayscale matrix. The resulting feature maps can therefore consist of simpler features, such as elementary parametrized curves, or Bézier curves with few control points. While the neural network can adapt to multiple artistic styles that can be found in a large dataset, the correspondence between the simple features and the neural representation remains to be questioned.**

***Index Terms*—hand-sketch, representation, inverse graphics, adversarial, RNN**

I. Introduction

Hand sketches are an ancient way for humans to communicate concepts through a simple representation that is intrinsically relatable. The analysis of the images of hand sketches raises questions such as how much the style can vary between artists and why.

Some potential applications of machine learning to the topic of hand sketch analysis are the possibility to study and discuss neural representations [1], generative models [2] and analysis of parametric curves for performing learning on compressed data [3, 4, 5].

A famous example of an end to end hand sketch learning platform is 'QuickDraw', an online automatic sketch recognition game which improves its results from the players input. The publicly available dataset has been subsequently cited in many other studies. The data has been further analysed to understand differences in artistic style based on the geographic location of the artist. [1] The ease of access to a very large crowdsourced dataset such as QuickDraw also makes it a potential candidate for studying the topic of synthetic data with the help of GAN [6] models.

There are also potential applications such as sketch based image retrieval [7], where the user intends to retrieve real pictures using a drawing.

In a fashion similar to the related problem of machine translation with RNN, CoSE (Compositional stroke embeddings) [8] uses the temporal ordering of the strokes for understanding drawings such as diagrams or 'complex free-form structures'.

Generative models are a popular case study in the context of hand-drawn sketches. An alternative to classical generative architectures are models such as DeepSVG [2], which use parametrized curves to produce animations.

In this paper we aim to discuss ***spline*** based stroke embeddings, including Bézier curves and other possible parametrized curves like NURBS (Non-uniform rational basis spline). The advantages of parametrized representations are the possibility of vector graphics generation, which is superior in some aspects to rasterized graphics.

II. Related work

A. Datasets

Aside from the 'QuickDraw' dataset which has 350 classes, there is the TU-Berlin dataset with 250 classes of objects that are sketched. CoSE [8] uses in addition to the QuickDraw dataset, the DiDi dataset (Digital Diagram Ink Data), in which special attention is payed to the temporal ordering of the strokes.[9]. For our focus, we make use mainly of the TU-Berlin dataset.

For analyzing the representation of these models with respect to compressed representations or linear transformations such as the Hough transform, only classification is used.

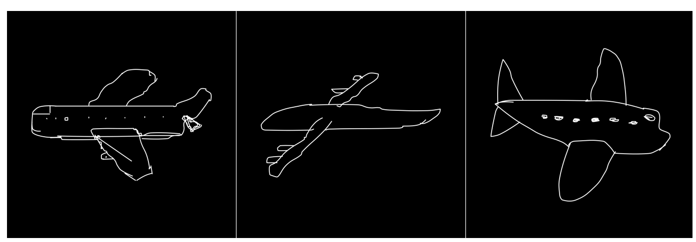

Fig. 1. First 3 images from the dataset, which might illustrate the intra-class variation.

B. SVM

HOG-SVM is a classical [1] bag of features based method. A bag of features histogram can be used to represent a sketch, after which a set of SVM classifiers is trained. [10].

HOG-SVM	56%
AlexNet-SVM	67.1%
MKL-SVM	65.8%
Sketch-a-Net	74.9%
Human	73.1%

[1]By classical, it is meant that advanced CNN models likely did not exist with good performance at the date of publishing in the context.

Table 1. A comparison of SVM methods that at the time did not surpass human accuracy.

C. CNN

The first success in hand sketch tasks was a CNN architecture named Sketch-a-Net [11], for the problem of classification, in 2015. The architecture consists of a CNN based deep learning model, and it was able to surpass the human benchmark of 73.1% accuracy for classification with a 74.9% accuracy score.

D. RNN, LSTM, GAN

The authors of 'QuickDraw' [1], introduced skech-rnn in their paper. Sketch-rnn is a sequence to sequence variational autoencoder, consisting of a bidirectional neural network. In addition to using generative models (GAN) on the task of sketch recognition, they also discuss how vectorized representations can be used in machine learning. Specifically they encode sketches into a latent space of embedding vectors, as well as analyzing stroke order. This illustrates that compressed representations can have other benefits.

E. Parametrized stroke embeddings

Splines or parametrized curves such as Bézier curves provide another way of embedding strokes. Using such approaches, stroke order can also be considered. The models learn a latent space for generating novel sketches, similar to other autoencoder architectures.

The generated sketches are shorter in terms of the number of points to be stored. The compact representation allows not only for simpler models for classification, but also interpolation on the latent space and other latent space operations using mathematical operations. [5] Another study achieves animating sketches by interpolation operations in the DeepSVG model. [2]

III. Methods

For achieving an intermediate representation between pixel-wise input data and NURBS representations of the same data, a Hough transform is applied on the data. The resulting map is a phase space picture of the sparse points with a common origin.

The Hough transform can be used to detect common shapes that might be found in the data, and is especially used for finding continuous segments of points such as lines or circles.

As part of the preprocessing, we invert the sketches for both version, to achieve a sparse matrix with the identity element 0 as predominant. This results in a small training advantage.

Most sketches appear in the probabilistic Hough transform as sine waves with points of intersection, which might appear to the neural network as features.

The transforms are used as inputs to the VGG-16 neural network, and trained with the Adam optimized and a learning rate of 1e-5. Instead of modifying the input shape to train on parametric curves, we still have an image as input to the CNN. This approach is chosen because we propose that even though

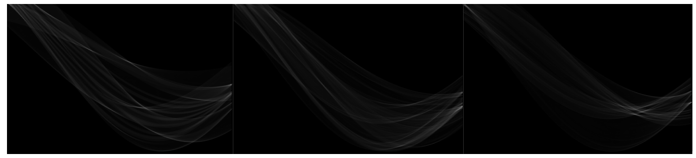

Fig. 2. The Hough transform of images from Fig. 1, in order

the Hough transform is a linear transform, the CNN might have a particular affinity for this representation, as the deep features attained by the network could be better suited for a phase space kind of input.

With a batch size of 2, the model converges to an approximation in less than 8 epochs. We then compare the classical learning model with the model using the transformed representations as input.

For further work we propose that further preprocessing the Hough transform into curve fitting representations, and lower dimensional representations, can increase computational efficiency, while reaching a similar evaluation performance.

Because of limited computational resources the classifier is restricted to the same 9 classes for both the classical and Hough transform versions.

IV. Results

The data is split into 50% train and 50% test data, with 9 classes of a total of 1500 images.

We reach within 8 epochs the training accuracy of 98%. The classic model has an eval accuracy of 0.69%, and the hough transform version has 0.79% accuracy.

Model	Accuracy	Precision	Recall
Classic VGG-16	0.6923	0.7105	0.6923
Hough VGG-16 (Ours)	0.7949	0.7949	0.7949

Table 2. Evaluation scores.

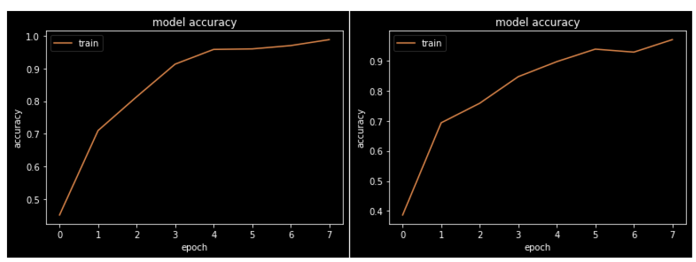

Fig. 3. The learning curves for Classic, respectively the Hough version

V. Discussion

Our suggested explanation for this 10% increase in accuracy of the Hough transform over the classical method is that intra-class variation is reduced when the transform is generated. The intra-class variation problem has been discussed in previous work, as referenced.

It could be possible that the number of features or convolutional filters that a classical neural network such as VGG-16 has is too small for the possible intra-class variation and noise of the hand drawing problem.

Because VGG-16's convolutional filters of the input might have overlapping regions over the sine wave data, a noise suppression effect could emerge, while the overlapping regions over noisy sparse matrices might have corner cases that increase the noise.

Bézier curves are a simple class of parametrized curves that could be used to extend this intermediate Hough phase portrait into a vectorized phase portrait. As previously mentioned, the curves are predominantly sinusoidal, so even simpler polynomial approximation might be used with success.

We suggest that further extending the compressed representation with novel architectures such as GNNs (Graph Neural Networks), or encoder-decoder architectures, could offer further possibilities for representation learning.

VI. Data and Code Availability

Experiments and code are available on GitHub: https://github.com/p0licat/cvdl-sketch-analysis

All datasets used are publicly available.

References

[1] David Ha and Douglas Eck. "A Neural Representation of Sketch Drawings". In: *CoRR* abs/1704.03477 (2017). arXiv: 1704.03477. URL: http://arxiv.org/abs/1704.03477.

[2] Alexandre Carlier et al. "DeepSVG: A Hierarchical Generative Network for Vector Graphics Animation". In: *CoRR* abs/2007.11301 (2020). arXiv: 2007.11301. URL: https://arxiv.org/abs/2007.11301.

[3] Jun Gao et al. "Deepspline: Data-driven reconstruction of parametric curves and surfaces". In: *arXiv preprint arXiv:1901.03781* (2019).

[4] Ayan Das et al. "BézierSketch: A generative model for scalable vector sketches". In: *CoRR* abs/2007.02190 (2020). arXiv: 2007.02190. URL: https://arxiv.org/abs/2007.02190.

[5] Ayan Das et al. "Cloud2Curve: Generation and Vectorization of Parametric Sketches". In: *CoRR* abs/2103.15536 (2021). arXiv: 2103.15536. URL: https://arxiv.org/abs/2103.15536.

[6] Ayan Kumar Bhunia et al. "Pixelor: A Competitive Sketching AI Agent. so You Think You Can Sketch?" In: *ACM Trans. Graph.* 39.6 (Nov. 2020). ISSN: 0730-0301. DOI: 10.1145/3414685.3417840. URL: https://doi.org/10.1145/3414685.3417840.

[7] Ayan Kumar Bhunia et al. "Sketch Less for More: On-the-Fly Fine-Grained Sketch Based Image Retrieval". In: *CoRR* abs/2002.10310 (2020). arXiv: 2002.10310. URL: https://arxiv.org/abs/2002.10310.

[8] Emre Aksan et al. "CoSE: Compositional Stroke Embeddings". In: *CoRR* abs/2006.09930 (2020). arXiv: 2006.09930. URL: https://arxiv.org/abs/2006.09930.

[9] Philippe Gervais et al. "The DIDI dataset: Digital Ink Diagram data". In: *CoRR* abs/2002.09303 (2020). arXiv: 2002.09303. URL: https://arxiv.org/abs/2002.09303.

[10] Yi Li, Yi-Zhe Song, and Shaogang Gong. "Sketch Recognition by Ensemble Matching of Structured Features". In: Jan. 2013, pp. 35.1–35.11. ISBN: 1-901725-49-9. DOI: 10.5244/C.27.35.

[11] Yongxin Yang and Timothy M. Hospedales. "Deep Neural Networks for Sketch Recognition". In: *CoRR* abs/1501.07873 (2015). arXiv: 1501.07873. URL: http://arxiv.org/abs/1501.07873.